

First Hit

End of Result Set



Generate Collection

Print

L3: Entry 1 of 1

File: PGPB

Jan 2, 2003

DOCUMENT-IDENTIFIER: US 20030004956 A1

TITLE: Methods and system for efficient association traversals

Abstract Paragraph:

Methods and systems are provided for performing efficient association traversal operations in a Common Information Model (CIM) implemented environment. A CIM Object Manager creates and defines instance objects and corresponding association objects in a repository. The CIM Object Manager creates wrappers for each object that include reverse links that show the relationship from an object to any corresponding association objects. When a relationship request for a selected object is received by the CIM Object Manager, it performs an association traversal operation using the reverse links stored in the repository. The CIM Object Manager locates the target object's wrapper in the repository, and collects relationship information from the reverse links included within the wrapper. The collected relationship information is then passed to the entity that initiated the request. Accordingly, efficient association traversals may be performed, without having to traverse each instance of defined association objects.

Series Code and Application Number:09/895077Summary of Invention Paragraph:

[0014] Methods, systems and articles of manufacture consistent with the present invention enable a CIM Object Manager (CIMOM) to create and store reverse links from an object to an association. The CIM Object Manager may then subsequently use the reverse links to efficiently determine the objects associated to a given input object, such as an instance of a class.

Summary of Invention Paragraph:

[0016] When the server node receives a client request for the relationship for a specific instance of a class, the CIM Object Manager accesses the reverse links stored in the wrapper of the instance associated with the request, or alternatively stored in the second level tables. The CIM Object Manager uses the reverse links to determine the relationship between the requested instance, and any other associated objects. The server node then sends the relationship information to the client.

Detail Description Paragraph:

[0036] In accordance with an aspect of the present invention, clients initiate CIM requests to create objects, generally, instances. The requests are received, via a network, at an object manager node incorporating a CIM Object Manager. The CIM Object Manager processes the requests and defines the objects stipulated in the client requests, and stores the definitions in a CIM Object Manager repository. Wrappers are then created for each defined object, and also are stored in the repository. The CIM Object Manager then defines associations related to the objects defined in the repository. For each association that is defined in the repository, the CIM Object Manager creates a reverse link that shows a reverse reference from each object to its respective associations. Thus, when a client initiates a relationship request associated with a particular object, the CIM Object Manager

---

utilizes the reverse links to process the request.

Detail Description Paragraph:

[0041] JVM 330 accepts requests from clients 320 for objects, including classes and S instances, defined in repository 350. These requests are interpreted and managed by CIM Object Manager 340. CIM Object Manager 340 is a CIM managing component that facilitates the interaction between a client 320 and management applications. CIM Object Manager 340 handles CIM based requests sent from clients 320, and manages repository 350. CIM Object Manager 340 defines classes, instances, associations, wrappers and reverse links.

Detail Description Paragraph:

[0042] Repository 350 is a storage area managed by CIM Object Manager 340. Repository 350 stores the definitions of classes, instances, associations, wrappers and reverse links created by CIM Object Manager 340.

Detail Description Paragraph:

[0046] Each wrapper 410 defines reverse links 440 (represented as dotted line arrows) that provide references from each instance 420, 450 to a respective association instance 430. For example, in FIG. 4, the association instance 430-1 shown between instances 420B and 420A receives reverse links 440-1, 440-2 that reference the relationships from instances 420B, 420A to the association instance 430-1, respectively. It should be noted that the objects shown in FIG. 4 are exemplary, and any number of objects and hierarchical relationships may be defined in the repository by CIM Object Manager 340.

Detail Description Paragraph:

[0047] CIM Object Manager 340 establishes and manages the definitions of instances 420, 450, associations instances 430, links 460, wrappers 410 and reverse links 440. Each time an instance is created in repository 350, CIM Object Manager 340 generates associations 430 based on the hierarchical relationship associated with each defined object. In turn, whenever an association is defined, CIM Object Manager generates reverse links within each respective wrapper. FIGS. 5A-5D illustrate this process, in accordance with one aspect of the invention.

Detail Description Paragraph:

[0049] FIG. 5B shows repository 350 containing the wrappers, or wrapper tables, associated with each instance illustrated in FIG. 5A. As shown, instance 5 wrapper table is related to wrapper 512 in FIG. 5A, while the instance wrapper tables I-4 correlate to the instance wrappers 523, 532, 542 and 552 in FIG. 5A, respectively. Each wrapper table includes a reference to an association instance that has been defined for its respective instance, depending upon the representation of the wrapper. That is, instance 5 wrapper table is defined for instance 510 (I-5) in FIG. 5A, and any associations defined for that instance will be represented in its wrapper table. Instance 5 wrapper table includes association 515, labeled A-1, corresponding to the relationship depicted in FIG. 5A. Also included in each wrapper table are reverse links established for each relationship with the association listed in a respective wrapper. Again referring to the instance 5 wrapper table, and FIG. 5A, it can be seen that association A-1 defines a relationship between instance I-5 of class 1 and instance I-1 of class 2. Accordingly, instance 5 wrapper table also defines the relationship of association A-1 with instance I-1. As there are no other associations defined for instance I-5, the remainder of its wrapper table is empty.

Detail Description Paragraph:

[0050] The remaining wrapper tables are similarly configured with information. Referring to instance 1 wrapper table and FIG. 5A, it can be seen that instance I-1 has four association instances related to it, A-1 through A-4. These are indicated in its wrapper table. Association instance A-1 has references to instance I-5, thus instance 1 wrapper table includes a reverse link that establishes this

relationship. The same is true for the relationships between association instances A-2 through A-4. Instance 1 wrapper table includes the reverse links that show the relationship between each association instance and instances I-2 through I-4.

Detail Description Paragraph:

[0052] FIG. 7 shows an exemplary process performed by CIM Object Manager 340 when a request to create a new instance is received from a client 320. A client 320 sends a request to create a new object, such as an instance of a particular class, over network 310 (Step 710). The request may be performed using XML/HTTP protocols. For example, one client 320 may initiate a request to create a new instance I-7 of class 1, while another client 320 may initiate a request to create a new instance I-6 of class 1. Referring to FIG. 5C, CIM Object Manager 350 receives the requests and defines the new instance I-7, and stores the definition in repository 350. CIM Object Manager 340 then creates a wrapper for the newly defined instance (Step 720). A wrapper is created by creating a wrapper table for the new instance. In this case, a new wrapper table for instance 7, would be created in repository 350. The new wrapper table includes default fields to be filled relating to associations and reverse links correlated to these associations. FIG. 5D shows repository 350 including newly defined wrappers, instance 6 wrapper table and instance 7 wrapper table.

Detail Description Paragraph:

[0053] CIM Object Manager 340 defines an association instance 564 (A-5) between instance I-5 and instance I-7 (step 730), using the characteristics of new instance I-7. FIG. 5C illustrates the new association 564 (A-5), including its links pointing to instances I-5 and I-7. Once CIM Object Manager 340 recognizes a newly defined association instance, it determines all objects the new association instance defines a relationship between. In this case, association instance A-5 establishes a relationship between instances I-5 and I-7. Accordingly, CIM Object Manager 340 defines reverse links within each instance's wrapper (Step 740). The reverse links are represented by the dotted lines pointing from wrappers 512 and 562 to association 564 (A-5) in FIG. 5C.

Detail Description Paragraph:

[0054] FIG. 5E illustrates how the wrappers in repository 350 are modified based on the created reverse links. The addition of newly defined instance I-7, and association A-5 triggers CIM Object Manager 340 to add the reverse links to the instance 5 wrapper table. As depicted, the instance 5 wrapper table includes a reverse link showing the relationship between association A-5 and instance I-7. Furthermore, newly defined wrapper table for instance I-7 is modified to include the relationship A-5 between instance I-7 and instance I-5.

Detail Description Paragraph:

[0055] The process for creating and maintaining the new instance wrapper for instance I-6 in repository 350 is the same as that described for instance I-7. That is, once CIM Object Manager 340 defines an association instance A-6 between instances I-6 and I-4, a new wrapper table for the new instance I-6 is created in repository 350 and reverse links (depicted in FIG. 5C as dotted lines pointing from wrappers 552 and 572 to association A-6) are defined within each instance's wrapper table. Referring to FIG. 5E, the newly defined instance 6 wrapper table includes a reverse link showing the relationship between association instance A-6 and instance I-4. Furthermore, instance 4 wrapper table includes a reverse link showing the relationship between association instance A-6 and instance I-6.

Detail Description Paragraph:

[0056] Accordingly, CIM Object Manager 340 dynamically adjusts repository 350 with new reverse links each time an instance of an association is created. This keeps repository 350 updated with the most recent relationships for instances defined by CIM Object Manager 340. In ill keeping an updated status of these relationships enable association traversals to be performed more efficiently by the CIM Object

Manager 340.

Detail Description Paragraph:

[0057] Although the wrapper table structure described above, provides efficient association traversals, the structure is not rigid. That is, other wrapper structures may be employed to provide traversal operations. In one aspect of the invention, the wrapper tables defined in repository 350, include pointers to secondary level association class wrapper tables. These secondary tables include each instance of a respective class association, and the reverse links to corresponding objects. FIGS. 6A-6E illustrate this aspect of the invention.

Detail Description Paragraph:

[0073] Moreover, it should be understood that the term "reverse link" is only a construct to aid in explaining features and principles of the present invention. Reverse links represent the relationships between objects and are not to be confused with actual physical "connections" between these objects.

CLAIMS:

1. A method for determining a relationship between objects related to a common information model, the objects including at least a first and second instance and an association, the method comprising: creating, for the first instance, a reverse link that defines a relationship between the first instance and the association; and determining a relationship between the first and second instances based on the reverse link.
3. The method of claim 1, wherein the instance is associated with a first wrapper defining the reverse link.
4. The method of claim 1, wherein the association represents an instance of an association class and wherein creating the reverse link further includes: defining a pointer in a first table that references a second table; and defining a pointer in the second table that references the instance of the association class.
6. A method for maintaining reverse links in a object-oriented environment including class instances and associations, the method comprising: for each class instance associated with N instances of an association class that each references the class instance, wherein N represents an integer value greater than or equal to one: (i) creating a first level wrapper table including a pointer to a second level wrapper table associated with the association class; and (ii) creating N pointers in the second level wrapper table that each references an individual instance of the association class.
9. A method for maintaining reverse links in an object-oriented environment including instances, association classes and association class instances, the method comprising: for each class instance associated with N association class that each includes at least one association class instance that references the class instance, wherein N represents an integer value greater than or equal to one: creating a first level wrapper table including N pointers to N second level wrapper tables associated with the association classes.
10. A method for maintaining reverse links in a object-oriented environment including class instances and associations, the method comprising: for each class instance associated with N instances of a first association class that each references the class instance, and X instances of a second association class that each reference the class instance, wherein N and X represent integer values greater than or equal to one: (i) creating a first level wrapper table including: a first pointer to a second level wrapper table associated with the first association class, and a second pointer to a second level wrapper table associated with the second association class; (ii) creating N pointers, in the second table wrapper

table associated with the first association class, that each references an individual instance of the first association class; and (iii) creating X pointers, in the second table wrapper table associated with the second association class, that each references an individual instance of the second association class.

21. A system for traversing associations in a common information model implemented environment, the model comprising at least a first and second instance and an association, comprising: means for creating, for the first instance, a reverse link that defines a relationship between the first instance and the association; and means for determining a relationship between the first and second instances based on the reverse link.

23. The system of claim 21, wherein the instance is associated with an first wrapper defining the reverse link.

24. The system of claim 21, wherein the association represents an instance of an association class and wherein the means for creating the reverse link further includes: means for defining a pointer in a first table that references a second table; and means for defining a pointer in the second table that references the instance of the association class.

25. The system of claim 24, wherein the means for determining a relationship includes: means for collecting a reference reflecting a relationship between the association and the second instance based on the pointer in the second table.

26 A system for maintaining reverse links in a object-oriented environment including class instances and associations, comprising: means for creating, for each class instance associated with N instances of an association class that each references the class instance, wherein N represents an integer value greater than or equal to one, a first level wrapper table including a pointer to a second level wrapper table associated with the association class; and means for creating N pointers in the second level wrapper table that each point to an individual instance of the association class.

37. A computer-readable medium including instructions for performing a method, when executed by a processor, for determining a relationship between objects related to a common information model, the objects including at least a first and second instance and an association, the method comprising: creating, for the first instance, a reverse link that defines a relationship between the first instance and the association; and determining a relationship between the first and second instances based on the reverse link.

39. The computer-readable medium of claim 37, wherein the instance is associated with an first wrapper defining the reverse link.

40. The computer-readable medium of claim 37, wherein the association represents an instance of an association class and wherein the step for creating the reverse link further includes: defining a pointer in a first table that references a second table; and defining a pointer in the second table that references the instance of the association class.

41. The computer-readable medium of claim 40, wherein the step of determining a relationship includes: collecting a reference reflecting a relationship between the association and the second instance based on the pointer in the second table.

42 A computer-readable medium including instructions for performing a method, when executed by a processor, for maintaining reverse links in a object-oriented environment including class instances and associations, the method comprising: for each class instance associated with N instances of an association class that each references the class instance, wherein N represents an integer value greater than

or equal to one: (i) creating a first level wrapper table including a pointer to a second level wrapper table associated with the association class; and (ii) creating N pointers in the second level wrapper table that each references an individual instance of the association class.

45. A computer-readable medium including instructions for performing a method, when executed by a processor, for maintaining reverse links in a object-oriented environment including class instances and associations, the method comprising: for each class instance associated with N association classes that each includes at least one association class instance that references the class instance, wherein N represents an integer value greater than or equal to one: creating a first level wrapper table including N pointers to N second level wrapper tables associated with the association classes.

46. A computer-readable medium including instructions for performing a method, when executed by a processor, for maintaining reverse links in a object-oriented environment including class instances and associations, the method comprising: for each class instance associated with N instances of a first association class that each references the class instance, and X instances of a second association class that each reference the class instance, wherein N and X represent integer values greater than or equal to one: (i) creating a first level wrapper table including: a first pointer to a second level wrapper table associated with the first association class, and a second pointer to a second level wrapper table associated with the second association class; (ii) creating N pointers, in the second table wrapper table associated with the first association class, that each references an individual instance of the first association class; and (iii) creating X pointers, in the second table wrapper table associated with the second association class, that each references an individual instance of the second association class.